Amendments to Claims

Please amend the claims as indicated in the listing below, which listing supercedes and replaces all prior listings of claims:

1. (Currently Amended) A method of operation of a control system, comprising

A. executing a first sequence of instructions in any of a first computer process and first computer thread, ~~(collectively, "first process")~~ collectively, "first process,

B. executing a second sequence of instructions in any of a second computer process and second computer thread, ~~(collectively, "second process")~~ collectively, "second process", the second process being loosely coupled with respect to the first process,

C. comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion by it of the second instruction sequence.

D. responding to failure of the first and second processes to achieve comparable states by rolling back each of the first and second processes to prior states in which a favorable comparison was achieved, and

~~E.~~ D. wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, a process control enterprise server, an industrial control device, an industrial control system, an environmental control device, an environmental control system, other control device, and other control system.

2. (Original) A method according to claim 1, comprising the step of executing step (C) one or more times over a time interval in order to determine whether the first and second processes achieve comparable states following completion of execution of the first instruction sequence by the first process.

3. (Original) A method according to claim 2, comprising the step responding to a favorable comparison in step (C) by repeating steps (A) – (C) with a third instruction sequence in place of

the first instruction sequence, and with a fourth instruction sequence in place of the second instruction sequence.

4. (Original) A method according to claim 3, comprising the steps of

selecting the third instruction sequence as a function of a state of the first process following execution of the first instruction sequence. and

selecting the fourth instruction sequence as a function of a state of the second process following execution of the second instruction sequence.

5. (Original) A method according to claim 4, comprising comparing a state of the first process prior to execution by it of the third instruction sequence with a state of the second process prior to execution by it of the fourth instruction sequence.

6. (Currently Amended) A method according to ~~any of~~ claims 1, 3 or ~~and~~ 5. comprising the step of signaling an error in response to a failure of the first and second processes to achieve comparable states at a time of comparison.

Claim 7 (cancelled).

8. (Original) A method according to claim 1, wherein step (C) comprises comparing any of registers, memory, flags, interrupts, tasks, and events in the respective processes.

9. (Original) A method according to claim 1, wherein each of the first and second processes comprise any of a thread and a process. and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

10. (Currently Amended) A control apparatus, comprising

A. any of a first computer process and first computer thread, ~~(collectively, "first process")~~ collectively, "first process, executing a first sequence of instructions.

B. any of a second computer process and second computer thread, ~~(collectively, "second process")~~ collectively, "second process", executing a second sequence of instructions,

C. comparison logic in communication with the first and second processes. the comparison logic comparing a state of the first process to a state of the second process following execution by them of the respective first and second sequences of instructions, the comparison logic signalling an error in response to a failure of the first and second processes to achieve comparable states at a time of comparison.

D. scheduling logic in communication with the first and second processes and in communication with the comparison logic, the scheduling logic responding to an error by rolling back each of the first and second processes to prior states in which a favorable comparison was achieved.

E.~~D.~~ wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server, industrial control device, an industrial control system, an environmental control device, an environmental control system, other control device, and other control system.

11. (Original) An apparatus according to claim 10, wherein the comparison logic compares any of registers, memory, flags, interrupts, tasks, and events in each of the respective processes.

12. (Original) An apparatus according to claim 11, wherein the comparison logic comprises

a first comparison logic section operating in the first process for comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion by it of the second instruction sequence, and

a second comparison logic section operating in the second process for comparing a state of the second process following completion by it of execution of the second instruction sequence with a state of the first process following completion, if any, by it of the first instruction sequence.

13. (Currently Amended) An apparatus according to claim 12, wherein ~~including scheduling logic in communication with the first and second processes and in communication with the comparison logic.~~

the scheduling logic schedules ~~scheduling~~ the first process to execute a third sequence of instructions in response to a favorable comparison by the comparison logic,

the scheduling logic schedules ~~scheduling~~ the second process to execute a fourth sequence of instructions in response to a favorable comparison by the comparison logic.

14. (Original) An apparatus according to claim 13, wherein the comparison logic compares a state of the first process prior to execution by it of the third instruction sequence with a state of the second process prior to execution by it of the fourth instruction sequence.

Claims 15 – 16 (cancelled).

17. (Original) An apparatus according to any of claims 10, 13 and 14, wherein the comparison logic compares a state of the first process with a state of the second process a plurality of times to determine whether the first and second processes to achieve comparable states.

18. (Original) An apparatus according to claim 10, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

19. (Currently Amended) A process control apparatus, comprising

A. any of a first computer process and a first computer task, ~~(collectively, "first process")~~ collectively, "first process,

B. any of a second computer process and a second computer task, ~~(collectively, "second process")~~ collectively, "second process",

C. scheduling logic in communication with the first and second processes, the scheduling logic scheduling the first process to execute a first subsequence of instructions from a first sequence of instructions, the scheduling logic scheduling the second process to execute a first subsequence of instructions from a second sequence of instructions, the scheduling logic responds to an error by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved,

D. comparison logic in communication with the first and second processes. the comparison logic comparing a state of the first process following completion by it of execution of the first instruction subsequence of instructions with a state of the second process following completion, if any, by it of the second subsequence of instruction, wherein states of the first and second processes include any of registers, memory, flags, interrupts, tasks, and events in those respective processes,

E. synchronization logic in communication with the comparison logic, the synchronization logic responding to a favorable comparison by the comparison logic by scheduling the first process to execute a second subsequence of instructions from the first sequence of instructions and scheduling the second process to execute a second subsequence of instructions from the second sequence of instructions.

20. (Original) An apparatus according to claim 19, wherein the scheduling logic comprises

a first scheduling logic section operating in the first process, the first scheduling logic section scheduling the first process to execute a first subsequence of instructions from a first sequence of instructions. and the first scheduling logic section responding to a favorable comparison by the comparison logic by scheduling the first process to execute a second subsequence of instructions from the first sequence of instructions, and

a second scheduling logic section operating in the second process, the second scheduling logic section scheduling the second process to execute a second subsequence of instructions from a first sequence of instructions, and the second scheduling logic section responding to a favorable comparison by the comparison logic by scheduling the second process to execute a second subsequence of instructions from the second sequence of instructions.

21. (Original) An apparatus according to claim 20, wherein the synchronization logic comprises

a first comparison logic section operating in the first process, the first comparison logic section comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion, if any, by it of the second instruction sequence, and

a second comparison logic section operating in the second process, the second comparison logic section comparing a state of the second process following completion by it of execu-

tion of the second instruction sequence with a state of the first process following completion. if any, by it of the first instruction sequence.

22. (Original) An apparatus according to claim 21. wherein at least one of the first and second comparison logic sections signals an error in response to a failure of the first and second processes to achieve comparable states following completion of execution of the respective subsequence of instructions by the respective process.

23. (Original) An apparatus according to claim 22, wherein the comparison logic compares a state of the first process with a state of the second process a plurality of times to determine whether the first and second processes to achieve comparable states.

24. (Original) An apparatus according to claim 22, wherein the scheduling logic responds to an error by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved.

25. (Original) An apparatus according to claim 19, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

26. (Original) An apparatus according to claim 19, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server.

27. (Currently Amended) A process control apparatus, comprising

A. any of a first computer process and a first computer task, ~~(collectively, "first process")~~ collectively, "first process, executing a first sequence of instructions.

B. any of a second computer process and a second computer task, ~~(collectively, "second process")~~ collectively, "second process", executing a second sequence of instructions,

C. synchronization logic in communication with the first and second processes, the synchronization logic preventing the first process from executing a third sequence of instructions until the first and second processes have completed execution of the first and second sequences of instructions. respectively,

D. at least one of the first and second comparison logic sections signals an error in response to a failure of the first and second processes to achieve comparable states following completion of execution of the respective subsequence of instructions by the respective process, and

E. the scheduling logic responds to an error by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved.

28. (Original) An apparatus according to claim 27, wherein the synchronization logic prevents the second process from executing a fourth sequence of instructions until the first and second processes have completed execution of the first and second sequences of instructions, respectively.

29. (Original) An apparatus according to any of claims 27 and 28, wherein the comparison logic compares a state of the first process with a state of the second process a plurality of times to determine whether the first and second processes to achieve comparable states.

30. (Original) An apparatus according to claim 27, wherein the synchronization logic comprises

a first comparison logic section operating in the first process, the first comparison logic section comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion, if any, by it of the second instruction sequence,

a second comparison logic section operating in the second process, the second comparison logic section comparing a state of the second process following completion by it of execution of the second instruction sequence with a state of the first process following completion, if any, by it of the first instruction sequence,

wherein states of the first and second processes include any of registers, memory, flags, interrupts, tasks, and events in those respective processes.

Claims 31 –32 (cancelled).

33. (Original) An apparatus according to claim 27, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

34. (Original) An apparatus according to claim 27, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server.

35. (Currently Amended) A method of process control, comprising

A. executing a first sequence of instructions in any of a first computer process and a first computer task ~~(collectively, "first process")~~, collectively, "first process."

B. executing a second sequence of instructions in any of a second computer process and a second computer task, ~~(collectively, "second process")~~, collectively, "second process," the second process being loosely coupled with respect to the first process,

C. comparing, within the first process, a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion, if any, by it of the second instruction sequence,

D. comparing, within the second process, a state of the second process following completion by it of execution of the second instruction sequence with a state of the first process following completion, if any, by it of the first instruction sequence,

E. wherein any of steps (C) and (D) include the step of signaling an error in response to a failure of the first and second processes to achieve comparable states following completion of execution of the respective instruction sequence by the respective process,

F. responding to such an error by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved.

36. (Original) A method according to claim 35, including the steps of

scheduling the first process to execute a third sequence of instructions in response to a favorable comparison in step (C), and

scheduling the second process to execute a fourth sequence of instructions in response to a favorable comparison in step (D).

Claim 37 (cancelled).

38. (Original) A method according to claim 37, wherein any of steps (C) and (D) include comparing a state of the respective process with a state of the other process a plurality of times to determine whether they achieve comparable states.

39. (Original) A method according to claim 35, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

40. (Original) A method according to claim 35, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server.

41. (Currently Amended) A method for process control, comprising

A. executing a first subsequence of instructions from a first sequence of instructions in any of a first computer process and a first computer task, collectively, "first process," ~~(collectively, "first process")~~,

B. executing a first subsequence of instructions from a second sequence of instructions in any of a second computer process and a second computer task, collectively, "second process," ~~(collectively, "second process").~~

C. comparing a state of the first process following completion by it of execution of the first subsequence of instructions from the first sequence of instructions with a state of the second process following completion, if any, by it of the first subsequence of instructions from the second sequence of instructions,

D. responding to a favorable comparison in step (C) by scheduling the first process to execute a second subsequence of instructions from the first sequence of instructions and scheduling the second process to execute a second subsequence of instructions from the second sequence of instructions,

E. responding to an unfavorable comparison in step (C) by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved.

42. (Original) A method according to claim 41, wherein step (C) includes

comparing, from within the first process, a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion, if any, by it of the second instruction sequence, and

comparing, from within the second process, a state of the second process following completion by it of execution of the second instruction sequence with a state of the first process following completion, if any, by it of the first instruction sequence.

Claim 43 (cancelled).

44. (Original) A method according to claim 43, wherein step (C) includes comparing a state of the first process with a state of the second process a plurality of times to determine whether the first and second processes to achieve comparable states.

45. (Original) A method according to claim 41, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

46. (Original) A method according to claim 41, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server.

47. (Currently Amended) A method of operation of a digital data processing system, comprising

A. executing a first sequence of instructions in any of a first computer process and first computer thread, ~~(collectively, "first process")~~ collectively, "first process",

B. executing a second sequence of instructions in any of a second computer process and second computer thread, ~~(collectively, "second process")~~collectively, "second process", the second process being loosely coupled with respect to the first process,

C. comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion by it of the second instruction sequence,

D. responding to failure of the first and second processes to achieve comparable states by rolling back each of the first and second processes to a prior states in which a favorable comparison was achieved

48. (Original) A method according to claim 47, comprising the step of executing step (C) one or more times over a time interval in order to determine whether the first and second processes achieve comparable states following completion of execution of the first instruction sequence by the first process.

49. (Original) A method according to claim 48, comprising the step responding to a favorable comparison in step (C) by repeating steps (A) (C) with a third instruction sequence in place of the first instruction sequence, and with a fourth instruction sequence in place of the second instruction sequence.

50. (Original) A method according to claim 49, comprising the steps of

selecting the third instruction sequence as a function of a state of the first process following execution of the first instruction sequence, and

selecting the fourth instruction sequence as a function of a state of the second process following execution of the second instruction sequence.

51. (Original) A method according to claim 50, comprising comparing a state of the first process prior to execution by it of the third instruction sequence with a state of the second process prior to execution by it of the fourth instruction sequence.

52. (Currently Amended) A method according to ~~any of~~ claims 47, 49 or ~~and~~ 51, comprising the step of signaling an error in response to a failure of the first and second processes to achieve comparable states at a time of comparison.

Claim 53 (cancelled).

54. (Original) A method according to claim 47, wherein step (C) comprises comparing any of registers, memory, flags, interrupts, tasks, and events in the respective processes.

55. (Original) A method according to claim 47, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

56. (Original) A method according to claim 47, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a process control plant server, and a process control enterprise server.

57. (Currently Amended) A digital data processing apparatus, comprising

A. any of a first computer process and first computer thread, ~~(collectively, "first process")~~ collectively, "first process," executing a first sequence of instructions,

B. any of a second computer process and second computer thread, ~~(collectively, "second process")~~ collectively, "second process", executing a second sequence of instructions,

C. comparison logic in communication with the first and second processes, the comparison logic comparing a state of the first process to a state of the second process following execution by them of the respective first and second sequences of instructions the comparison logic signalling an error in response to a failure of the first and second processes to achieve comparable states at a time of comparison,

D. scheduling logic in communication with the first and second processes and in communication with the comparison logic, the scheduling logic responding to an error by rolling back each of the first and second processes to prior states in which a favorable comparison was achieved.

58. (Original) An apparatus according to claim 57, wherein the comparison logic compares any of registers, memory, flags, interrupts, tasks, and events in each of the respective processes.

59. (Original) An apparatus according to claim 58, wherein the comparison logic comprises

a first comparison logic section operating in the first process for comparing a state of the first process following completion by it of execution of the first instruction sequence with a state of the second process following completion by it of the second instruction sequence, and

a second comparison logic section operating in the second process for comparing a state of the second process following completion by it of execution of the second instruction sequence with a state of the first process following completion, if any, by it of the first instruction sequence.

60. (Original) An apparatus according to claim 59, including scheduling logic in communication with the first and second processes and in communication with the comparison logic,

the scheduling logic scheduling the first process to execute a third sequence of instructions in response to a favorable comparison by the comparison logic,

the scheduling logic scheduling the second process to execute a fourth sequence of instructions in response to a favorable comparison by the comparison logic.

61. (Original) An apparatus according to claim 60, wherein the comparison logic compares a state of the first process prior to execution by it of the third instruction sequence with a state of the second process prior to execution by it of the fourth instruction sequence.

Claims 62 –63 (cancelled).

64. (Original) An apparatus according to any of claims 57, 60 and 63, wherein the comparison logic compares a state of the first process with a state of the second process a plurality of times to determine whether the first and second processes to achieve comparable states.

65. (Original) An apparatus according to claim 57, wherein each of the first and second processes comprise any of a thread and a process, and wherein the first and second processes execute on any of the same and different digital data processing apparatus.

66. (Original) An apparatus according to claim 57, comprising wherein each of the first and second processes execute on any of a process control field device, a block controller, a process controller, a proc ss control plant server, and a process control enterprise server.